GLOMATION

Embedded Single Board Computer

GESBC-9G10
User's Manual

Table of Contents

Chapter 1 – Introducing the GESBC-9G10 Single Board Computer 4
GESBC-9G10 Overview 4
Advanced Features 4
AT91SAM9G10 5
SDRAM 5
FLASH 5
USB 5
UART 1 5
DEBUG Port 5
Ethernet 6
Chapter 2 – Getting Started 7
Assembly and Connections 7
Operation 7
Configurations 9
Chapter 3 – GESBC-9G10 Function Blocks 10
AT91SAM9G10 10
SDRAM 10
FLASH 10
USB 11
RS-232 Port 0 and 1 11
Ethernet 11
USB Port 11
GPIO 12
LCD Interface 13
RTC 14
JTAG 14
Power Requirement 14
Chapter 4 – Software Description 15
Overview 15
Data Storage on GESBC-9G10 15
GESBC-9G10 Linux Code 15
U-boot 15
U-boot Booting Linux 15
Loading Linux Kernel and root File System 16
Chapter 5 – Development Tools 18
Overview 18
Linux Development Tool Chain 18
Chapter 6 – Troubleshooting 20

List of Tables

Table 1 System Configuration 9
Table 2 UART Port P0 Connector on GESBC-9G10 11
Table 3 UART2 Port 1 Connector 11
Table 4 J17 USB Device Port 12
Table 5 J16 I/O Expansion 12
Table 6 J9 LCD interface 13
Table 7 J20 JTAG Connector 14
Table 8 J1 Power Supply Connector 14
Table 9 NAND FLASH Storage Map 15

Chapter 1 – Introducing the GESBC-9G10 Single Board Computer

GESBC-9G10 Overview

The GESBC-9G10 is a low cost compact sized single board computer based on Atmel AT91SAM9G10 processor. With a large peripheral set targeted to a variety of applications, the GESBC-9G10 is well suited for industrial controls, digital media servers, audio jukeboxes, thin clients, set-top boxes, point-of-sale terminals, biometric security systems, and GPS devices.

Advanced Features

The heart of the GESBC-9G10 is the AT91SAM9G10 which is the one in a series of ARM926EJ-S-based processors. The AT91SAM9G10 microcontroller features DSP Instruction Extensions, ARM Jazelle® Technology for Java® Acceleration. It has separate 32 Kbyte instruction and data caches with write buffer. The ARM926EJ-S on the AT91SAM9G10 functions with a maximum operating clock rate of 266MHz and a power usage between 20mW and 80mW (dependent upon clock speed). The ARM core operates from a 1.2V supply while the I/O operates at 3.3V. The low power consumption makes it an idea platform for battery operated applications.

The list below summarizes the features of the GESBC-9G10.

- 266MHz Processor Core – ARM926EJ-S with MMU
- 64~ 128 MB SDRAM
- 256MB ~ 1GB NAND FLASH
- 1 10/100 Mbps Ethernet port
- 2 RS-232 Universal Asynchronous Receiver / Transmitters (UARTs)
- 2 USB Host Port
- 1 USB Device Port
- Real-Time Clock with battery backup
- Hardware Debug Interface
- SD/MMC Socket
- On-chip watchdog timer
- LCD supports Passive or Active Displays, up to 16-bits per pixel in STN Color Mode, up to 16M Colors in TFT Mode (24-bit per Pixel), resolution up to 2048 x 2048

Figure 1 below shows a picture of the GESBC-9G10 Single Board Computer.

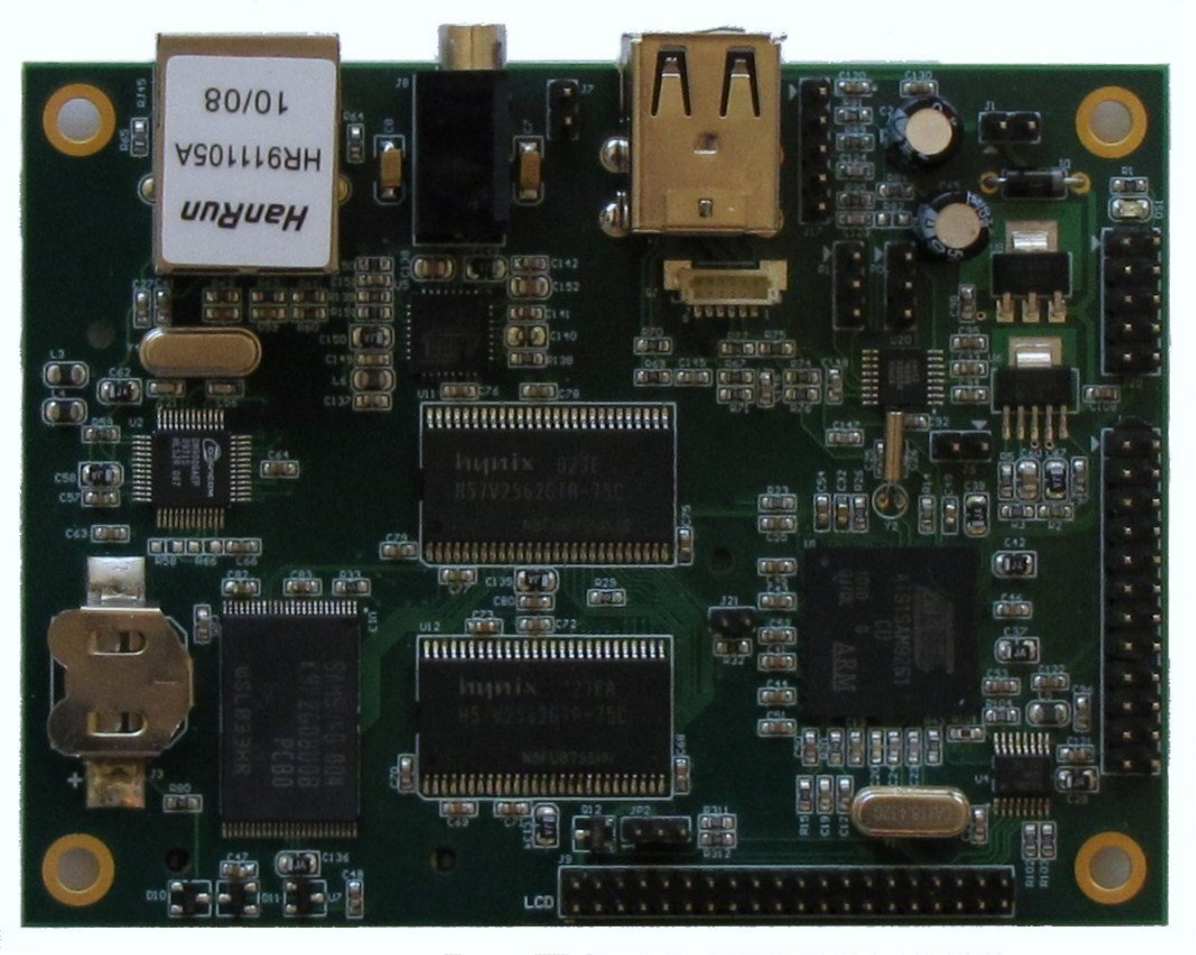


Figure 1. GESBC-9G10 Single Board Computer

AT91SAM9G10

The GESBC-9G10 is shipped with the Atmel AT91SAM9G10 processor. For more information regarding the AT91SAM9G10 processor please see the AT91SAM9G10 datasheet.

SDRAM

The GESBC-9G10 is shipped with 64MBytes of SDRAM.

FLASH

The GESBC-9G10 is shipped with 256MB NAND FLASH.

USB

The GESBC-9G10 is shipped with two USB host ports.

UART 1

The GESBC-9G10 is shipped with the 3 wire UART 1 interface.

DEBUG Port

The GESBC-9G10 is shipped with the 3 wire serial debug port.

Ethernet

The GESBC-9G10 is shipped with a 100-base TX Ethernet with integrated status LEDs.

Audio Output

The GESBC-9G10 is shipped with a Atmel stereo audio DAC AT73C213 chip with 32 Ohm/20 mW stereo headset output

Chapter 2 – Getting Started

This chapter describes the GESBC-9G10 working environment and familiarizes the user with its components and functionality. This chapter contains the following sections:

- Assembly and Connections
    - Describes how to assemble and connect components to the GESBC-9G10 Single Board Computer

- Operation
    - Describes how to operate the GESBC-9G10 Single Board Computer

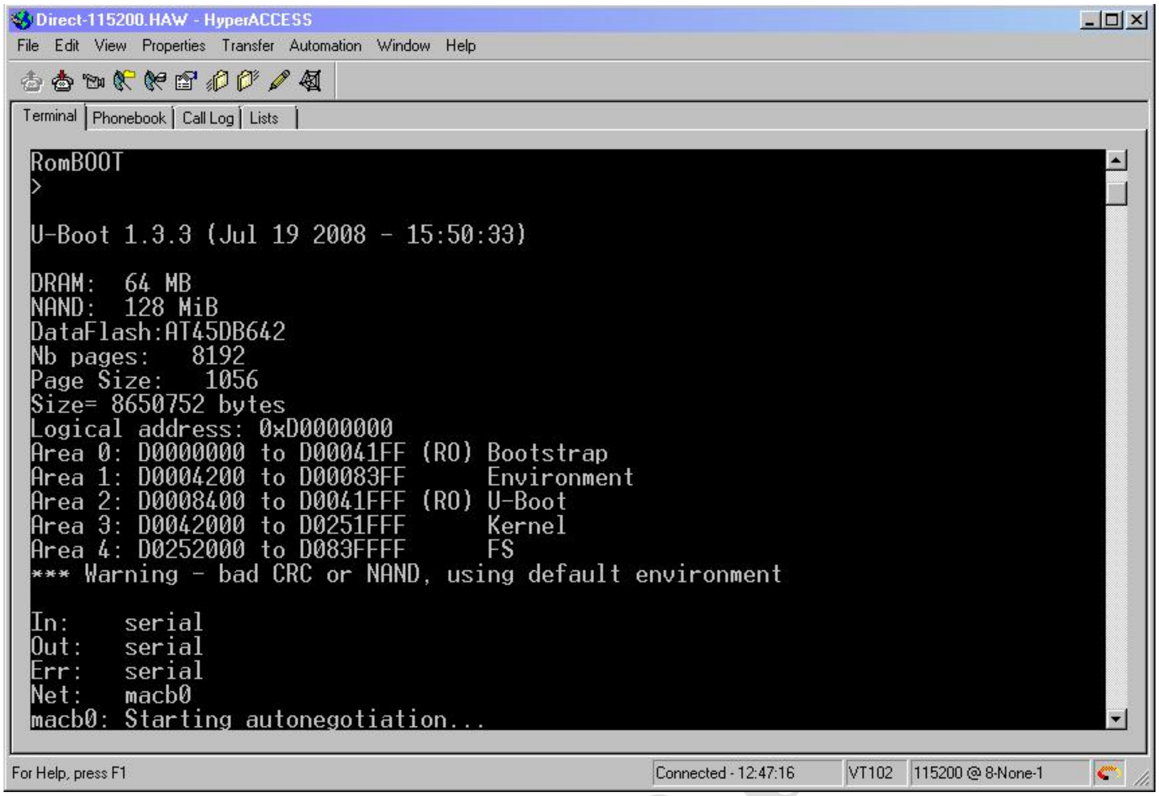
Assembly and Connections

In order to use the GESBC-9G10 the user must first assemble and connect the peripherals to the GESBC-9G10, as described in the following procedure.

1. Place the GESBC-9G10 on a static free surface.
2. Connect 5V regulated power supply to the board.
3. Connect null modem serial cable between GESBC-9G10 debug port and PC/terminal serial port.
4. Launch a terminal emulator, such as HyperTerminal, or minicom, on the PC configured to connect to the serial port of the GESBC-9G10. Configure the serial port with the following parameters: 115200 bits per second, 8 data bits, no parity, 1 stop bit, no flow control.
5. Connect the board to a local area network (optional)

Operation

A few seconds after applying power to the GESBC-9G10, debug information will be displayed on the terminal program. The following figures show what this should look like.



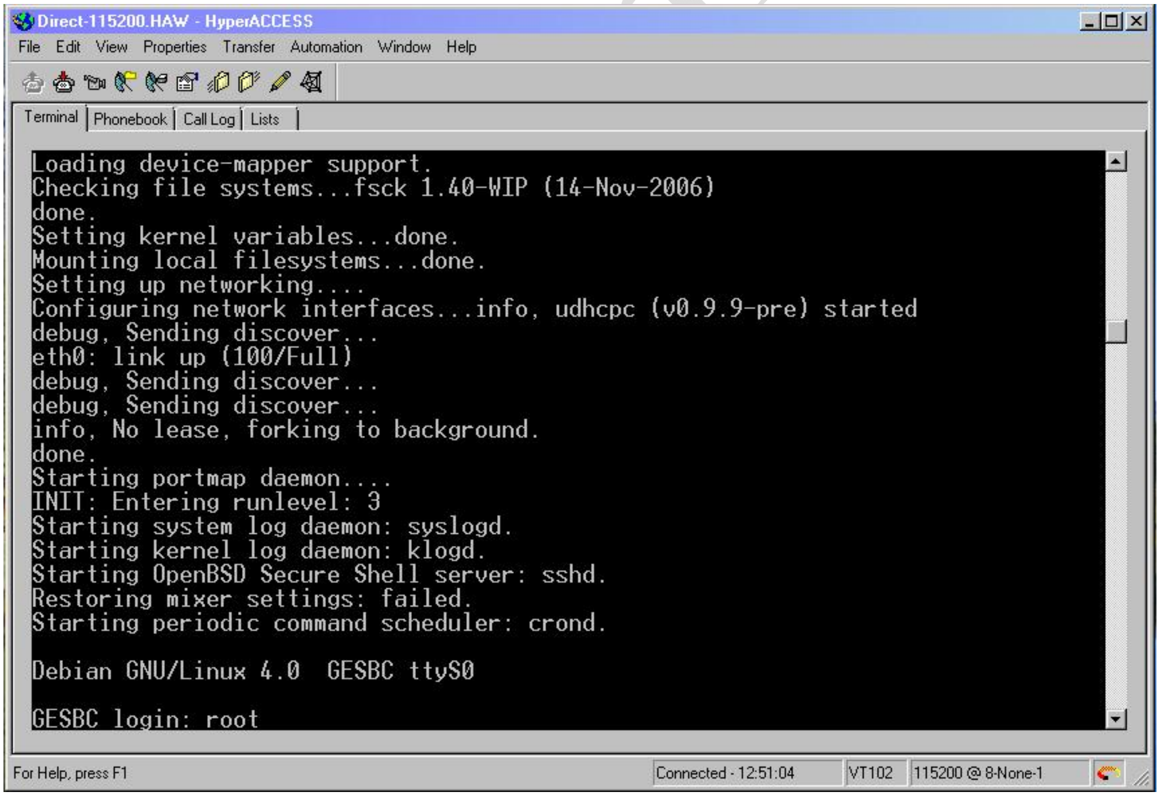
Direct-115200.HAW - HyperACCESS
File Edit View Properties Transfer Automation Window Help
Terminal Phonebook Call Log Lists
For Help, press F1
Connected - 12:47:16
VT102
115200 @ 8-None-1

```
RomBOOT
>

U-Boot 1.3.3 (Jul 19 2008 - 15:50:33)

DRAM:  64 MB
NAND:  128 MiB
DataFlash:AT45DB642
Nb pages:   8192
Page Size:   1056
Size= 8650752 bytes
Logical address: 0xD0000000
Area 0: D0000000 to D00041FF (RO) Bootstrap
Area 1: D0004200 to D00083FF      Environment
Area 2: D0008400 to D0041FFF (RO) U-Boot
Area 3: D0042000 to D0251FFF      Kernel
Area 4: D0252000 to D083FFFF      FS
*** Warning - bad CRC or NAND, using default environment

In:    serial
Out:   serial
Err:   serial
Net:   macb0
macb0: Starting autonegotiation...
```



Direct-115200.HAW - HyperACCESS
File Edit View Properties Transfer Automation Window Help
Terminal Phonebook Call Log Lists
For Help, press F1
Connected - 12:51:04
VT102
115200 @ 8-None-1

```
Loading device-mapper support.
Checking file systems...fsck 1.40-WIP (14-Nov-2006)
done.
Setting kernel variables...done.
Mounting local filesystems...done.
Setting up networking....
Configuring network interfaces...info, udhcpc (v0.9.9-pre) started
debug, Sending discover...
eth0: link up (100/Full)
debug, Sending discover...
debug, Sending discover...
info, No lease, forking to background.
done.
Starting portmap daemon....
INIT: Entering runlevel: 3
Starting system log daemon: syslogd.
Starting kernel log daemon: klogd.
Starting OpenBSD Secure Shell server: sshd.
Restoring mixer settings: failed.
Starting periodic command scheduler: crond.

Debian GNU/Linux 4.0  GESBC ttyS0

GESBC login: root
```

Please see

Chapter 4 – Software Description for more details regarding the software functionality.

Configurations

Jumpers are used to configure the GESBC-9G10 to operate in different mode. The following table lists all the settings for each jumper.

Table 1 System Configuration

Jumper	Description
BP1	System reset switch header

Chapter 3 – GESBC-9G10 Function Blocks

AT91SAM9G10

The GESBC-9G10 Single Board Computer uses the Atmel AT91SAM9G10 as the core processor on this development board. The top-level features of AT91SAM9G10 processor are the following:

- ARM926EJ-S RISC Core Processor
- 266 MHz / 266 MIPS Performance
- 16Kbyte Instruction Cache
- 16Kbyte Data Cache
- Linux and Windows CE enabled MMU
- 100 MHz System Bus
- 32 bit SDRAM Interface
- 32 bit SRAM / FLASH / ROM Interface
- Serial EEPROM Interface
- 3 UART
- Two-port USB Host
- 2 SPI Port
- Serial Audio Interface
- JTAG Interface

More detailed information regarding the AT91SAM9G10 processor can be found at www.atmel.com.

SDRAM

The AT91SAM9G10 features a unified memory address model where all memory devices are accessed over a common address and data bus. The GESBC-9G10 supports up to 128MB SDRAM.

FLASH

The GESBC-9G10 is shipped with 256 Mbytes of NAND FLASH memory. The GESBC-9G10 can be also ordered with optional 512MB ~ 1GB NAND FLASH.

USB

The GESBC-9G10 Single Board Computer provides two USB host connections. The AT91SAM9G10 USB host controller is configured for two root hub ports and features an integrated transceiver for each port. The AT91SAM9G10 integrates two USB 2.0 Full Speed host ports. These ports are fully compliant to the OHCI USB 2.0 Full Speed specification (12 Mbps). The controller complies with the OHCI specification for USB Revision 1.1. The USB ports are brought out by a standard double deck USB type A connector.

The GESBC-9G10 Single Board Computer provides one USB device port. The USB Device Port (UDP) is compliant with the Universal Serial Bus (USB) V2.0 full-speed device specification.

RS-232 Port 0 and 1

The GESBC-9G10 Single Board Computer is shipped with two 3-wire RS-232 UART interface.

The port 0 is the debug USART port of the AT91SAM9G10. The P0 connector is the 3 pin header on GESBC-9G10. The signal designation is listed in the following tables.

Table 2 UART Port P0 Connector on GESBC-9G10

Pin Number	Signal Name
1	RX
2	TX
3	GND

The serial port 1 is the USART 0 on the AT91SAM9G10 processor. It is provided via the 3 pin header on GESBC-9G10. The signal designation is listed in the following tables.

Table 3 UART2 Port 1 Connector

Pin Number	Signal Name
1	RX
2	TX
3	GND

Ethernet

The GESBC-9G10 Single Board Computer is shipped with support for a complete 10/100Mbps Ethernet interface.

USB Port

The GESBC-9G10 Single Board Computer is shipped with 2 USB host port on standard USB type-A double deck connector.

The GESBC-9G10 Single Board Computer is shipped with one USB device port J17. The USB device port signal assignment is listed in the following table.

Table 4 J17 USB Device Port

Pin Number	Signal Name
1	USB connection
2	DM
3	DP
4	GROUND
5	GROUND

GPIO

The AT91SAM9G10 contains very rich set of peripherals that are multiplex into 2 groups, Peripheral A and Peripheral B, with individually programmable pins. The SPI bus, A/D and GPIO are provided together with other functions on the I/O expansion port. The I/O expansion port is a 2x12 2.54mm spacing male header. The following table lists signals available on the I/O expansion connector with their corresponding multiplexed functions and default usage on the GESBC-9G10 Single Board Computer.

Table 5 J16 I/O Expansion

Pin	I/O Line	Peripheral A	Peripheral B	Comments	Function
1					+3.3V
2					+3.3V
3	PA7	TWD	PCK0		
4	PA8	TWCK	PCK1		
5	PA12	TCLK	RTS1		
6	PA13	TPS0	CTS1		
7	PA14	TPS1	SCK2		
8	PA15	TPS2	RTS2		
9	PA20	TPK4	RD1		
10	PA21	TPK5	RK1		
11	PA22	TPK6	RF1		
12	PA23	TPK7	RTS0		
13	PA24	TPK8	SPI1_NPCS1		
14	PA25	TPK9	SPI1_NPCS2		
15	PA26	TPK10	SPI1_NPCS3		
16	PA27	TPK11	SPI0_NPCS1		
17	PA30	TPK14	A23		
18	PA31	TPL15	A24		
19	PC6	CFCE1			
20	PC7	CFCE2			
21	PC12	TXD1	NCS6		
22	PC13	RXD1	NCS7		

23					GND
24					GND

For more detailed information on multiplexed peripherals please see AT91SAM9G10 data sheet.

LCD Interface

The GESBC-9G10 provides a LCD interface that supports passive or active LCD displays, up to 16-bits per pixel in STN Color Mode, up to 16M Colors in TFT Mode (24-bit per Pixel), resolution up to 2048 x 2048. The LCD interface also supports 4-wire resistive touch screen. The J9 is a 40 pin 2mm spacing male header. The following table lists the LCD interface signals.

Table 6 J9 LCD interface

Pin	Function
1	VCC-lcd
2	VCC-lcd
3	LCDD0
4	LCDD1
5	LCDD2
6	LCDD3
7	LCDD4
8	LCDD5
9	LCDD6
10	LCDD7
11	LCDD8
12	LCDD9
13	LCDD10
14	LCDD11
15	LCDD12
16	LCDD13
17	LCDD14
18	LCDD15
19	LCDD16
20	LCDD17
21	LCDD18
22	LCDD19
23	LCDD20
24	LCDD21
25	LCDD22
26	LCDD23
27	LCDCC
28	PB30
29	NC
30	LCDDOTCK
31	LCDVSYNC

32	LCDDEN
33	LCDHSYNC
34	PA16
35	XP
36	YP
37	XM
38	YM
39	GND
40	GND

RTC

The GESBC-9G10 uses the AT91SAM9G10 on-chip RTC with battery hook-up to provide accurate time keeping.   The on-board battery holder accepts CR1225/CR1220 coin cell batteries.

JTAG

The GESBC-9G10 Single Board Computer is shipped with a 10 pin connector that provides JTAG debug signals for the CPU. The JTAG provides the user with the ability to debug system level programs. The signal designation is listed in the following table.

Table 7 J20 JTAG Connector

Pin Number	Signal Name	Pin Number	Signal Name
1	3.3V	2	3.3V
3	NTRST	4	TDI
5	TMS	6	TCK
7	RTCK	8	TDO
9	GND	10	GND

Power Requirement

The GESBC-9G10 Single Board Computer requires regulated 5V DC. The power supply should have minimum 400mA capacity for the GESBC-9G10 board itself. Additional capacity may be required if a LCD is connected to the GESBC-9G10 and power is fed through the LCD interface..

Table 8 J1 Power Supply Connector

Pin Number	Signal Name
1	5V DC
2	GND

Chapter 4 – Software Description

Overview

This chapter provides information regarding the software that is shipped with the GESBC-9G10 Board. The software included with the board is U-boot boot loader, Linux kernel 2.6.25, and Debian distribution style compact root file system. The applications included provide access to all hardware functions on the GESBC-9G10 board.

Data Storage on GESBC-9G10

The default configuration of the GESBC-9G10 Single Board Computer uses on board NAND FLASH for all data storage requirements, including boot strap code, boot loader, Linux kernel, and Linux file system.

The following table is the factory default storage map on the NAND FLASH.

Table 9 NAND FLASH Storage Map

Start Address	Size	Usage
0x00000000	0x20000	Boot strap code
0x00020000	0x40000	U-boot
0x00060000	0x20000	U-boot primary environment storage range
0x00080000	0x20000	U-boot secondary environment storage range
0x00100000	0x300000	Linux kernel
0x00400000	--	Root file system

GESBC-9G10 Linux Code

The GESBC-9G10 is shipped with Linux 2.6.30 kernel pre-installed. This software is programmed into the system FLASH located on the board prior to shipment. The Linux kernel is configured with all the device drivers included for the GESBC-9G10 board.

U-boot

U-boot provides a simple interface for loading operating systems and applications onto the GESBC-9G10 board. U-Boot uses a serial console for its input and output. The default serial port setting is 115200,8,N,1. It also supports the built-in Ethernet port and general flash programming.

The board is shipped with U-boot pre-installed. Please refer to U-boot user's manual regarding detailed information of U-boot.

U-boot Booting Linux

The following shows the default U-boot setup for booting Linux.

```
U-Boot> printenv
bootargs=console=ttyS0,115200 root=/dev/mtdblock2 rw rootfstype
=jffs2 mtdparts=atmel_nand:1M(bootloader),3M(kernel),-(rootfs)
bootcmd=nand read.jffs2 0x22000000 0x100000 0x200000; bootm
bootdelay=1
baudrate=115200
ethaddr=00:0c:20:02:0a:5b
ipaddr=192.168.0.200
serverip=192.168.0.102
netmask=255.255.255.0
stdin=serial
stdout=serial
stderr=serial
ethact=macb0

Environment size: 353/131067 bytes
U-Boot>
```

The `bootcmd` setting of the U-boot reads the Linux kernel from NAND FLASH at address 0x100000 to SDRAM at address 0x22000000 and start executing the kernel code at the same memory address. The NAND FLASH from 0x400000 and up is used for Linux root file system. The U-boot passes the MTD device partition setting to the Linux kernel via the `bootargs` environment variable.

Loading Linux Kernel and root File System

The U-boot boot-loader provides many ways to load Linux kernel and file system into FLASH memory. The loading by Ethernet network is shown here. User can consult U-boot manual for other methods of loading data.

After power on the GESBC-9G10 board, stop the U-boot auto-execution by press any key. The following message should be shown on the terminal console on the host PC connected to the GESBC-9G10 board.

```
RomBOOT
>

U-Boot 1.3.3 (Jul 19 2008 - 15:50:33)

DRAM:  64 MB
NAND:  256 MiB
In:    serial
Out:   serial
Err:   serial
Net:   macb0
macb0: Starting autonegotiation...
macb0: Autonegotiation timed out (status=0x7849)
macb0: link up, 100Mbps full-duplex (lpa: 0x4de1)
Hit any key to stop autoboot:  0
U-Boot>
```

The network address and server address must be set before network transfer can take place. The following commands will set the SBC IP address and server IP address,

```
set ipaddr xxx.xxx.xxx.xxx
set serverip xxx.xxx.xxx.xxx
```

The server IP is the IP address where a TFTP server must be run. To load Linux kernel type in the following command,

```
t 0x22000000 uImage
```

The U-boot will load uImage file from the TFTP server whose IP address is specified by the `serverip` environment variable.

The NAND FLASH sectors must be erased first before new kernel image can be stored. The following command will erase the NAND FLASH sectors reserved for Linux kernel,

```
nand erase 0x100000 0x200000
```

The use the flowing command to store the kernel image from SDRAM to NAND FLASH,

```
nand write.jffs2 0x22000000 0x100000 0x200000
```

The following commands can be used to load root file system into the FLASH memory,

```
nand erase 0x400000 [available_nand_flash_memory_size]
t 0x21000000 rootfs.img
nand write.jffs2 0x21000000 x0400000 $(filesize)
```

Please be noted that the image is first loaded into the SDRAM and then stored into the FLASH memory. The image size can not exceed the available SDRAM on the board.

After the kernel and root file system have been updated the board can be simply reboot by recycle the power.

Chapter 5 – Development Tools

Overview

This chapter provides a brief introduction to development tools that are available for the AT91SAM9G10 System-on-a-Chip processor. The central processing core on the AT91SAM9G10 is a 200 MHz ARM926EJ-S processor. The ARM926EJ-S RISC processing core is supported through various toolsets available from third party suppliers. The typical toolset required for the code development is a compiler, assembler, linker and a source-level code debugger. Code debugging is supported via the on-chip JTAG interface.

Linux Development Tool Chain

The Linux development tool chain is available at Glomation website in the support page. A host PC running Linux operating system is required to run the development tools. This guide assumes user had basic Linux or Unix application development knowledge.

Host Computer Requirement

The host PC should run Redhead, SuSe, or other Linux distribution, a RS-232 serial port, at least 500MB free disk space, and a terminal program such as minicom.

Hardware Connection

A null modem cable is required to connect GESBC-9G10 to the host computer.

Install Linux Development Tool Chain

The ARM Linux Development Tool chain can be installed in any directory on the host system. The following example uses cross compiler default directory /usr/local/arm as the installing directory for the ARM Linux cross compiler.

1. Login as root and untar the tool chain

   ```
   cd /
   tar jxvf /<cross compiler tar file directory>/ Generic-arm_gcc-4.2.3-
   glibc-2.3.3.tar.bz2
   ```

2. Set up the directory path variable

   ```
   export PATH=/usr/local/arm/gcc-4.2.3-glibc-2.3.3/arm-unknown-linux-
   gnu/bin:$PATH
   ```

above command can be included in the shell resource file so it is executed every time you login. For bash shell, a good place to put is in .bashrc in your home directory.

Compile Linux Kernel

The GESBC-9G10 is shipped with Linux kernel version 2.6.30. The patch for the kernel source tree is available at http://www.linux4sam.org/twiki/bin/view/Linux4SAM/LinuxKernel.

Prepare Linux Kernel source

Obtain the kernel source 2.6.30 from http://www.kernel.org. Untar the Linux kernel,

```
tar xjf  linue-2.6.30.bz2
```

Patch the kernel source with patches for Atmel AT91SAM9G10-EK,

```
patch –p1 < /<patch-file-directory-path>/patch_file_name
```

Configure Linux Kernel

The GESBC-9G10 can use the default configuration file for the Atmel AT91SAM9G10-ek evaluation board.

```
make ARCH=arm CROSS_COMPILE=arm-unknown-linux-gnu-
AT91SAM9G10ek_defconfig
```

If additional configuration is required, executing the following command in the Linux kernel directory,

```
make ARCH=arm CROSS_COMPILE=arm-unknown-linux-gnu- menuconfig
```

If problem occurs, make sure the default PATH variable is set to the correct tool chain directory

Compile Kernel

Once Linux kernel has been configured, it can be compiled using following command[1],

```
make ARCH=arm CROSS_COMPILE=arm-unknown-linux-gnu- uImage
```

The Linux kernel should compile without error and the image file will be created.

[1] The U-boot tool mkimage must be pre-installed in order to make final uImage.

Chapter 6 – Troubleshooting

This chapter provides Troubleshooting information. Search the entries in the Problem column in order to find the item that best describes your situation. Then perform the corrective action in the same row. If the problem persists, contact Glomation.